Efficient Fine-Tuning of Image-Conditional Diffusion Models for Depth and Surface Normal Estimation

Gonzalo Martin Garcia[1] **Karim Abou Zeid**[1] **Christian Schmidt**[1]

Daan de Geus[1,2] **Alexander Hermans**[1] **Bastian Leibe**[1]

[1]RWTH Aachen University [2]Eindhoven University of Technology

https://vision.rwth-aachen.de/diffusion-e2e-ft

Abstract

Recent work showed that large diffusion models can be reused as highly precise monocular depth estimators by casting depth estimation as an image-conditional image generation task. While the proposed model achieved state-of-the-art results, high computational demands due to multi-step inference limited its use in many scenarios. We show that the inefficiency was caused by a flaw in the inference pipeline that has so far gone unnoticed. The fixed model performs comparably to the best previously reported configurations while being more than 200× faster. Furthermore, we show that end-to-end finetuning with task-specific losses enables deterministic single-step inference, outperforming previous diffusion-based depth and normal estimation models on common zero-shot benchmarks. This fine-tuning scheme works similarly well on Stable Diffusion directly.

1 Introduction

Recent work has reused large diffusion models [29] for monocular depth estimation by casting depth prediction as a conditional image generation task [19]. The resulting models show good task performance and remarkably high levels of details. However, the consensus in the community is that they tend to be slow [19, 11, 13], due to many evaluations of a large neural network during inference.

We argue that, contrary to common belief, such image-conditional diffusion models [19, 11] should yield reasonable predictions with a single inference step. We investigate the behavior of Marigold [19], an image-conditional diffusion-based depth estimator, and find dismal performance in the few-step regime due to a critical flaw in the DDIM implementation. We demonstrate that this flaw is particularly critical in the scope of image-conditional methods such as Marigold.

With a small correction to the inference pipeline, Marigold-like models [19, 11] obtain single-step performance that is comparable to multi-step, ensembled inference, while **being more than 200× faster**. In fact, this bug-fix makes diffusion-based depth estimators speed-wise comparable to state-of-the-art discriminative depth estimation models, opening up exciting avenues for further improvements, such as more efficient fine-tuning and self-training with pseudo-labels [39, 40].

We fine-tune Marigold end-to-end into a deterministic affine-invariant monocular depth estimator using a scale and shift invariant loss function [27]. This model outperforms the best configurations of Marigold, producing sharp and accurate outputs in a single step. We repeat this experiment with the task of surface normal estimation and find similar results: end-to-end fine-tuning with a task-specific loss outperforms more complicated architectures which were trained on more data. Following Occam's Razor, we find that even the simplest baseline, direct fine-tuning of Stable Diffusion (SD) [29] into a deterministic feed-forward model, outperforms Marigold and other diffusion-based depth- and normal estimation methods.

We demonstrate the effectiveness of this fine-tuning scheme on common zero-shot benchmarks. Our deterministic single-step model outperforms other diffusion-based depth- and normal estimation

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

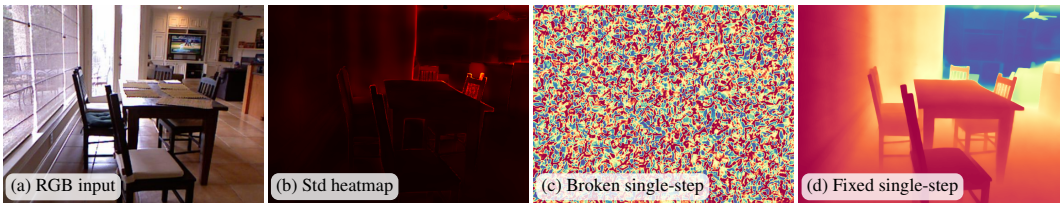

Figure 1: **Marigold output visualizations.** (a) RGB input; (b) pixel-wise standard deviation of Marigold's depth map output during 50-step DDIM inference; and Marigold's depth map prediction (c) before and (d) after fixing the inference pipeline.

methods, and achieves results comparable to state-of-the-art methods for affine-invariant depth prediction and surface normal estimation.

2 Related Work

Monocular Geometry Estimation. Monocular depth estimation involves predicting a pixel-wise depth map of a scene from a single image. A commonly used representation is *affine-invariant depth*, which refers to metric depth up to an unknown global scale and shift. This representation is widely adopted in various monocular depth estimation methods [43, 27, 26, 8, 18, 39, 40, 19, 13]. Surface normal estimation, which predicts surface orientations in the form of pixel-wise 3D vectors, has been addressed both prior to the rise of deep learning [15, 16] and in the deep learning era [37, 9, 3, 1, 8, 18, 2, 17]. Recently, many methods leverage powerful pretrained backbones [39, 40, 41, 17, 25, 29] and evaluate generalization to "in-the-wild" scenes and zero-shot benchmarks.

Diffusion Models for Geometry Estimation. Several works have explored the use of diffusion models for depth estimation [32, 31, 7, 44], but these models have not demonstrated robust generalization. More recently, Marigold [19] fine-tuned Stable Diffusion [29] to produce detailed and accurate depth maps, conditioned on RGB images. The core idea behind this approach is that SD's ability to model realistic images also provides strong geometric and semantic priors, essential for accurate depth estimation. Marigold's success is attributed to its training on high-quality synthetic datasets with perfect ground truth, and a smooth transition in the latent space from text-conditioned images to image-conditioned depth, thereby preserving the model's generalization capability.

Marigold has inspired several follow-up works. DiffCalib [14], for instance, extends Marigold by jointly predicting depth and camera intrinsics through the addition of an incident map [45], which is denoised along with the depth map. GeoWizard [11] predicts both depth and surface normals using two parallel UNet evaluations with cross-attention between the two branches. However, both models suffer from high computational costs during inference due to multi-step denoising. DepthFM [13] combines Marigold's core ideas with Flow Matching [22] to reduce the number of inference steps while preserving high output quality. Additionally, the authors of Marigold now provide an LCM-distilled [24] version that allows for few-step evaluation, albeit at a reduced quality.

In our work, we observe that Marigold and its derivatives, aside from DepthFM and Marigold LCM, which are designed for few-step prediction, suffer from a flawed implementation [21] of the DDIM [35] inference pipeline that prevents them from functioning effectively in the few-step regime. Moreover, while the denoising diffusion fine-tuning objective used by Marigold and its successors has been effective, we find that it is not a key factor for the models' success.

3 Depth and Normal Estimation with Image-Conditional Diffusion Models

Marigold. Marigold adapts a pretrained Stable Diffusion [29] UNet for depth estimation. As a latent diffusion model, SD operates in the latent space of a Variational Autoencoder (VAE); Marigold uses this VAE to encode RGB images and depth maps into latent representations. Similar to SD, Marigold does not train the VAE, but only the UNet. The training task is to denoise the depth map latents given the (clean) RGB latents. To this end, the RGB latents and noised depth map latents are concatenated to form the input, and the first convolutional layer of the UNet is duplicated. Marigold uses the $\mathbf{v}$-parametrization objective [30] and multi-resolution noise [38] during training. The training dataset consists of a small set of high-quality synthetic images with perfect ground-truth depth maps.

During inference, the model samples a depth map latent matching the latent of the input RGB image, and the (frozen) VAE decoder decodes this latent into the predicted depth map. Marigold uses the

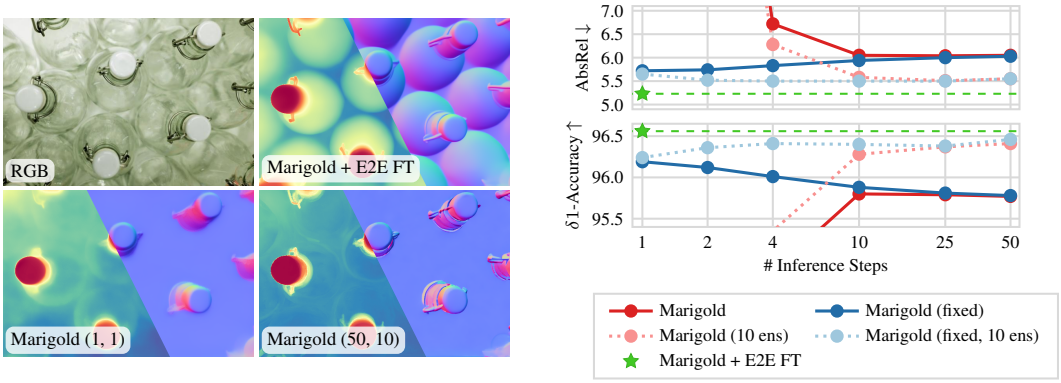

Figure 2: **Qualitative samples (left) and performance comparison of Marigold variants (right).** The end-to-end fine-tuned version outperforms both single-step and ensembled multi-step Marigold.

DDIM [35] inference scheduler, which allows to sample with fewer steps than the model was trained with (*e.g.*, Marigold trains with $T = 1000$ diffusion steps, but samples with 10 to 50 steps). The default implementation of this scheduler yields bad single-step inference results (see Fig. 1c).

Fixing Single-Step Inference. We show the pixel-wise standard deviation across steps for Marigold's default 50-step inference in Fig. 1b. We observe values close to zero for almost all pixels, indicating that the prediction changes only slightly during the 50-step schedule. However, the single-step output corresponds to pure noise, as seen in Fig. 1c.

We find that this discrepancy is caused by a flaw in the DDIM scheduler implementation used by Marigold and some derivative works [19, 11]. The flaw causes the model to receive an inconsistent pairing of timestep and noise; in particular, for a single-step prediction, the model receives a timestep encoding that indicates an almost perfect depth map ($t = 1$) whereas the actual input is pure noise. The model receives significantly more noise than it expects, and forwards the noise almost unchanged.

Fixing the flaw is simple: we need to align the timestep with the noise level, starting inference at $t = T$. To do this, we can use the `trailing` setting as proposed in recent work [21] for image-generative models. However, we emphasize that while this setting provided only slight improvements for image generation [21], this setting is crucial for single-step inference in models such as Marigold, as shown in Fig. 1 (and Fig. 5 in the appendix).

End-to-End Fine-Tuning. With the fixed inference pipeline, Marigold shows good performance even in the single-step inference setting. However, the denoising objective does not optimize the task of interest and produces either slightly blurry outputs (with single-step inference) or over-sharpened outputs (with multi-step inference); see Fig. 2 (left) and the supp. material for qualitative samples. Additionally, the best configurations still use ensembles, as seen in Fig. 2 (right). To fix this, we directly fine-tune the UNet end-to-end, using a task-specific loss function (affine-invariant [27] for depth estimation and an angular loss for surface normal estimation; see the supp. material for details).

We continue to train Marigold for single-step inference, using $t = T$ and replacing the noise with the mean of the noise distribution, *i.e.*, zeros. According to the $\mathbf{v}$-parametrization [30], this setting corresponds to single-step prediction. The output of the UNet is then decoded using the frozen VAE decoder and compared to the ground-truth depth map. Note the difference between this fine-tuning approach and Marigold's diffusion fine-tuning objective: Marigold trains to match the *latents* of the GT depth maps using an MSE loss; instead, we optimize to predict good *decoded depth maps*.

4 Experimental Evaluation

Experimental Setup. To enable a direct comparison with Marigold [19], we use the same training hyperparameters and evaluation protocol, with the only significant differences being the loss functions and fixing the timestep to $t = T$. Importantly, we also keep the same training datasets: Hypersim [28] and Virtual KITTI 2 [5]. For depth estimation, we use an affine-invariant loss function [27]; for normal estimation, the loss is based on the angle between the ground truth and predicted normals.

Improved Inference Results. As is evident from Fig. 2, the fixed DDIM scheduler reveals that Marigold's [19] multi-step denoising is not actually working: instead of improving the depth map

Table 1: **Depth estimation results.**

Method	Steps	Ens.	Inference time	NYUv2 [34]		KITTI [12]		ETH3D [33]		ScanNet [6]		DIODE [36]	
				AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑
Marigold [19]	50	10	24 s	5.5	96.4	9.9	91.6	6.5	**96.0**	6.4	95.1	30.8	77.3
Marigold [19]	50	1	3.1 s	6.0	95.9	10.5	90.4	7.1	95.1	6.9	94.5	31.0	77.2
Marigold LCM	4	5	1.8 s	6.2	95.6	9.9	91.7	6.9	95.5	7.0	94.5	30.9	77.6
Marigold LCM	1	1	**121** ms	6.5	95.4	10.7	89.9	7.5	94.5	7.6	93.8	31.5	76.3
Marigold + DDIM fix	1	1	**121** ms	5.7	96.2	10.8	89.6	6.9	95.5	6.6	95.2	31.1	76.8
Marigold + E2E FT	1	1	**121** ms	**5.2**	**96.6**	**9.6**	91.9	**6.2**	95.9	**5.8**	96.2	**30.2**	**77.9**
SD [29] + E2E FT	1	1	**121** ms	5.4	96.5	**9.6**	**92.1**	6.4	95.9	**5.8**	**96.5**	30.3	77.6

Table 2: **Normal estimation results.**

Method	Steps	Ens.	Inference time	NYUv2 [34]		ScanNet [6]		iBims-1 [20]		Sintel [4]	
				Mean↓	11.25°↑	Mean↓	11.25°↑	Mean↓	11.25°↑	Mean↓	11.25°↑
Marigold [19]	50	10	24 s	18.8	55.9	17.7	58.8	18.4	64.3	39.1	14.9
Marigold + DDIM fix	1	1	**121** ms	17.4	56.5	16.8	57.6	18.1	62.9	37.1	15.7
Marigold + E2E FT	1	1	**121** ms	**16.2**	**61.4**	**14.7**	66.0	**15.8**	**69.9**	**33.5**	21.5
SD [29] + E2E FT	1	1	**121** ms	16.5	60.4	**14.7**	**66.1**	16.1	69.7	**33.5**	**22.3**

with more denoising steps, the performance actually *degrades*. This behavior was previously masked by the large error due to the broken DDIM scheduler; the fixed model performs strictly better than before, for any given number of steps. Ensembling does still provide benefits when using at least two inference steps. For single-step inference, the predictions tend to be highly correlated, in which case ensembling does not lead to significant improvements. Single-step performance of the fixed model is comparable to the best configuration originally reported for Marigold.

Comparison with Diffusion-Based Baselines. We compare Marigold and a variant distilled into a Latent Consistency Model (LCM) [24] for few-step inference against our single-step variants in Tab. 1. Marigold LCM reduces the number of inference steps and the ensemble size compared to vanilla Marigold, but performs slightly worse. The single-step model with fixed DDIM scheduler yields better performance than LCM distillation—*without additional training*. End-to-end fine-tuning of this model further improves the performance, achieving single-step results that outperform the best configuration of vanilla Marigold (50 steps, 10 ensemble). Surprisingly, direct fine-tuning of Stable Diffusion [29] (*i.e.*, without Marigold's diffusion training stage) yields comparable performance.

Tab. 2 presents similar findings for surface normal estimation: The fixed single-step Marigold outperforms the multi-step ensembled version, and end-to-end fine-tuning yields even better results, even when starting directly from Stable Diffusion. We observed the same behavior for the more complex GeoWizard [11]; the results of those experiments can be found in the supp. material.

5 Conclusion

We have shown that a critical flaw in the implementation of the DDIM scheduler severely degrades single-step inference performance in prior image-conditional diffusion-based depth estimators. This flaw caused several prior works to draw wrong conclusions. We find that diffusion-based depth estimation does not need to be slow and indeed works with single-step inference; multi-step inference does not improve predictions; the denoising training objective is not key for depth estimation and is outperformed by simple end-to-end fine-tuning. Nonetheless, our work supports the hypothesis that diffusion pretraining does provide excellent priors for geometric tasks such as monocular depth and normal estimation. The resulting models allow accurate single-step inference, making sophisticated self-training procedures [39, 40] feasible on large-scale data. We believe that further improvements in diffusion models will lead to even more reliable priors, which might further improve the performance of this kind of geometry estimation models.

Acknowledgments and Disclosure of Funding

Karim Abou Zeid's research is funded by the Bosch-RWTH LHC project "Context Understanding for Autonomous Systems". Christian Schmidt was funded by BMBF project bridgingAI (16DHBKI023). Computations were performed with computing resources granted by RWTH Aachen University under project rwth1690.

References

[1] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *ICCV*, 2021. 2

[2] Gwangbin Bae and Andrew J Davison. Rethinking inductive biases for surface normal estimation. In *CVPR*, 2024. 2, 11

[3] Aayush Bansal, Bryan C. Russell, and Abhinav Kumar Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. *CVPR*, 2016. 2

[4] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 4, 10, 11, 12

[5] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 3, 9, 10

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 4, 10, 11, 12

[7] Yiqun Duan, Xianda Guo, and Zheng Zhu. DiffusionDepth: Diffusion denoising approach for monocular depth estimation. *arXiv preprint arXiv:2303.05021*, 2023. 2

[8] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *ICCV*, 2021. 2, 11

[9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2

[10] Yi Feng, Bohuan Xue, Ming Liu, Qijun Chen, and Rui Fan. D2NT: A high-performing depth-to-normal translator. In *ICRA*, 2023. 8, 10

[11] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. *ECCV*, 2024. 1, 2, 3, 4, 10, 11, 12

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 4, 10, 11, 12

[13] Ming Gui, Johannes S. Fischer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast monocular depth estimation with flow matching. *arXiv preprint arXiv:2403.13788*, 2024. 1, 2, 10, 11, 12

[14] Xiankang He, Guangkai Xu, Bo Zhang, Hao Chen, Ying Cui, and Dongyan Guo. Diffcalib: Reformulating monocular camera calibration as diffusion-based dense incident map generation. *arXiv preprint arXiv: 2405.15619*, 2024. 2

[15] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. *SIGGRAPH*, 2005. 2

[16] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Recovering surface layout from an image. *IJCV*, 2007. 2

[17] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE TPAMI*, 2024. 2, 10, 11

[18] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *CVPR*, 2022. 2, 11

[19] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024. 1, 2, 3, 4, 7, 8, 9, 11, 12

[20] Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of cnn-based single-image depth estimation methods. In *ECCVW*, 2018. 4, 10, 11, 12

[21] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. *WACV*, 2023. 2, 3, 7

[22] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 2

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 10

[24] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 2, 4

[25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[26] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 2, 11

[27] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE TPAMI*, 2020. 1, 2, 3, 8, 10, 11

[28] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 3, 9, 10

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 4, 7, 8, 10, 11, 12

[30] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 2, 3, 7

[31] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *NeurIPS*, 2023. 2

[32] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023. 2

[33] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 4, 10, 11, 12

[34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 4, 10, 11, 12

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021. 2, 3

[36] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 4, 10, 11, 12

[37] X. Wang, David F. Fouhey, and Abhinav Kumar Gupta. Designing deep networks for surface normal estimation. *CVPR*, 2015. 2

[38] Jonathan Whitaker. Multi-resolution noise for diffusion model training. https://wandb.ai/johnowhitaker/multires_noise/reports/Multi-Resolution-Noise-for-Diffusion-Model-Training--VmlldzozNjYyOTU2 (2024-09-09), 2023. 2, 7

[39] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 1, 2, 4, 10, 11

[40] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 1, 2, 4, 10, 11

[41] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *ICCV*, 2023. 2, 10, 11

[42] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *CVPR*, 2021. 11

[43] Chi Zhang, Wei Yin, Zhibin Wang, Gang Yu, Bin Fu, and Chunhua Shen. Hierarchical normalization for robust monocular depth estimation. *NeurIPS*, 2022. 2, 11

[44] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023. 2

[45] Shengjie Zhu, Abhinav Kumar, Masa Hu, and Xiaoming Liu. Tame a wild camera: In-the-wild monocular camera calibration. In *NeurIPS*, 2023. 2

Efficient Fine-Tuning of Image-Conditional Diffusion Models for Depth and Surface Normal Estimation

Supplemental Material

A Marigold Diffusion Training

Fig. 3 shows an overview of Marigold's training procedure. Marigold adapts the SD v2 [29] UNet architecture for conditional depth map generation, using $\mathbf{v}$-*parametrization* [30] during training. The training objective is formulated in the latent space of the frozen SD VAE. The GT depth map $\mathbf{d}^*$ is replicated along the channel dimension to conform to the 3-channel inputs of the VAE and encoded as $\mathbf{z}^* = \mathcal{E}(\mathbf{d}^*)$, where $\bar{\alpha}_t = \prod_{t=1}^{T} \alpha_t$ and α_t corresponds to the noise schedule. Similarly, the RGB latent is $\mathbf{x} = \mathcal{E}(\mathbf{I}_{\text{RGB}})$. During training, noise is added only to the depth latent to get $\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}^* + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$. The UNet receives the concatenated latents and the timestep t as inputs and predicts $\hat{\mathbf{v}}_t = \hat{\mathbf{v}}_\theta([\mathbf{z}_t, \mathbf{x}], t)$. The first convolutional layer is duplicated to accomodate the larger number of input channels [19].

The optimization target $\mathbf{v}_t^*$ at timestep t is then a linear combination of the sampled noise $\boldsymbol{\epsilon}$ and the GT depth map latent $\mathbf{z}^*$ such that $\mathbf{v}_t^* = \sqrt{\bar{\alpha}_t}\boldsymbol{\epsilon} - \sqrt{1-\bar{\alpha}_t}\mathbf{z}^*$. The model is optimized with a squared error objective comparing the model prediction $\hat{\mathbf{v}}_t$ with $\mathbf{v}_t^*$. With the chosen noise schedule, $t = 0$ corresponds to little noise in the input, and the model is forced to predict the noise; nearer to $t = T$, the input is mostly noise and the model should predict a denoised image. Additionally, the authors observed significantly improved depth estimation performance by training with annealed multi-resolution noise [38] and sampling with isotropic Gaussian noise [19].

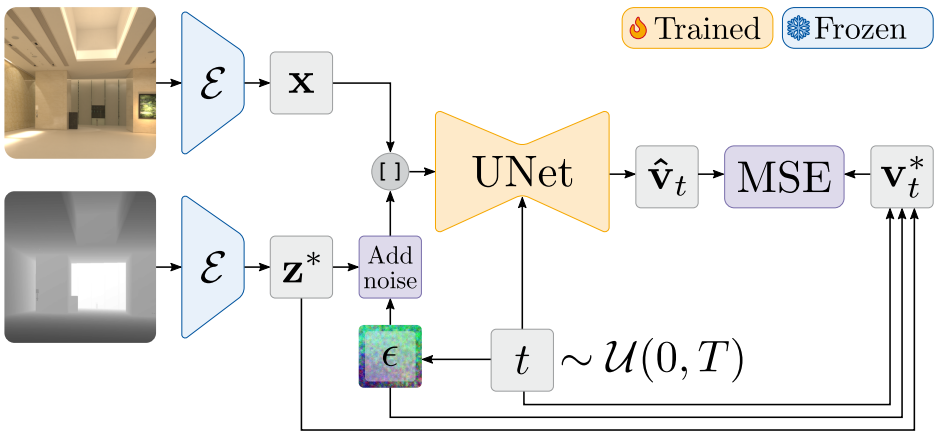

Figure 3: **Marigold [19] diffusion training for conditional depth map generation.** Marigold starts with a pretrained Stable Diffusion v2 model [29], which is fine-tuned for image-conditional generation of depth maps or surface normal maps.

B DDIM Inference

During training, the highest noise level corresponds to the last timestep $t = T$, and $t = 1$ corresponds to a very small noise level. The DDIM inference scheduler iterates over a series of k timesteps $\tau_1 > \tau_2 > \ldots > \tau_k > 0$ and iteratively denoises the initial noise input $\mathbf{z}_{\tau_1}$. We consider the `leading` and `trailing` schedules that are also discussed by Lin *et al.* [21] and show the selected timesteps for different k in Tab. 3. The original `leading` timestep selection strategy of the DDIM scheduler excludes the final timestep T. This leads to a mismatch between training and inference; using the `leading` schedule, the model expects a partially denoised input in the first step, but instead receives only noise. In contrast, the fixed `trailing` strategy always starts with $t = T$ for the first denoising step, properly aligning training and inference. In the limit of $k \to T$ inference steps, both strategies converge to the same behavior.

Table 3: **Comparison of `leading` vs. `trailing` timestep selection.** The timesteps selected by two DDIM scheduler timestep selection strategies for $T = 1000$ timesteps and varying numbers of inference steps.

Inference Steps	`leading` timestep selection	`trailing` timestep selection
1	[1]	[999]
2	[501, 1]	[999, 499]
4	[751, 501, 251, 1]	[999, 749, 499, 249]
10	[901, 801, 701, 601, 501, 401, 301, 201, 101, 1]	[999, 899, 799, 699, 599, 499, 399, 299, 199, 99]

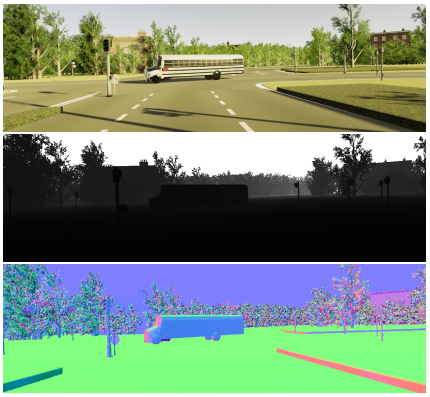

Figure 4: **Virtual KITTI example.** Top: Synthetic RGB image. Middle: Ground-truth depth map. Bottom: Ground-truth surface normals, generated using discontinuity-aware gradient filters [10].

Figure 5: **Single-step outputs of Marigold and Stable Diffusion.** With a single step, Stable Diffusion produces a blurry image at best, while Marigold outputs a sensible depth map. Note that the input prompt is text for Stable Diffusion, but an RGB image for Marigold.

In Fig. 5, we illustrate the difference between single-step predictions using the broken `leading` and the fixed `trailing` DDIM scheduler for Marigold [19] and Stable Diffusion [29]. Both models output noise when using the broken scheduler. With the fixed implementation, both models predict the mean of their respective conditional distribution. For single-step Marigold this results in a well-defined depth map, whereas for single-step Stable Diffusion, it produces a blurry image with coarse structures that roughly align with the input prompt.

Fig. 6 further demonstrates the scheduler's impact when multiple steps are considered. It clearly shows that the effect of the broken scheduler becomes less noticeable as the number of inference steps increases. Additionally, the weak text conditioning in Stable Diffusion leads to blurry images, which gradually sharpen as more inference steps are taken. In contrast, the strong image conditioning in Marigold allows the model to predict reasonably accurate depth maps already in the first step. As shown by the heatmap in Fig. 1b, subsequent steps only lead to small changes in the predicted distances, and most of the scene remains unchanged.

C Detailed Experimental Setup

Task-specific Losses for End-to-End Training. For monocular depth estimation, we use an affine-invariant loss function [27] which is invariant to global scale and shift of the depth map. In particular, we perform least-squares fitting between the ground-truth depth $\mathbf{d}^*$ and the predicted depth map $\mathbf{d}$ to estimate the scale and shift values s and t. The aligned prediction is then given as $\hat{\mathbf{d}} = s\mathbf{d} + t$, and the loss function is defined as

$$\mathcal{L}_{\mathrm{D}} = \frac{1}{HW} \sum_{i,j} \left| d^*_{i,j} - \hat{d}_{i,j} \right| , \tag{1}$$

where (i, j) denotes the pixel coordinates, and H and W are the height and width of the image, respectively.

For surface normal estimation, we use a loss based on the angle between the ground truth and predicted normals:

$$\mathcal{L}_{\mathrm{N}} = \frac{1}{HW} \sum_{i,j} \arccos \left(\frac{n^*_{i,j} \cdot \hat{n}_{i,j}}{\|n^*_{i,j}\| \|\hat{n}_{i,j}\|} \right) , \tag{2}$$

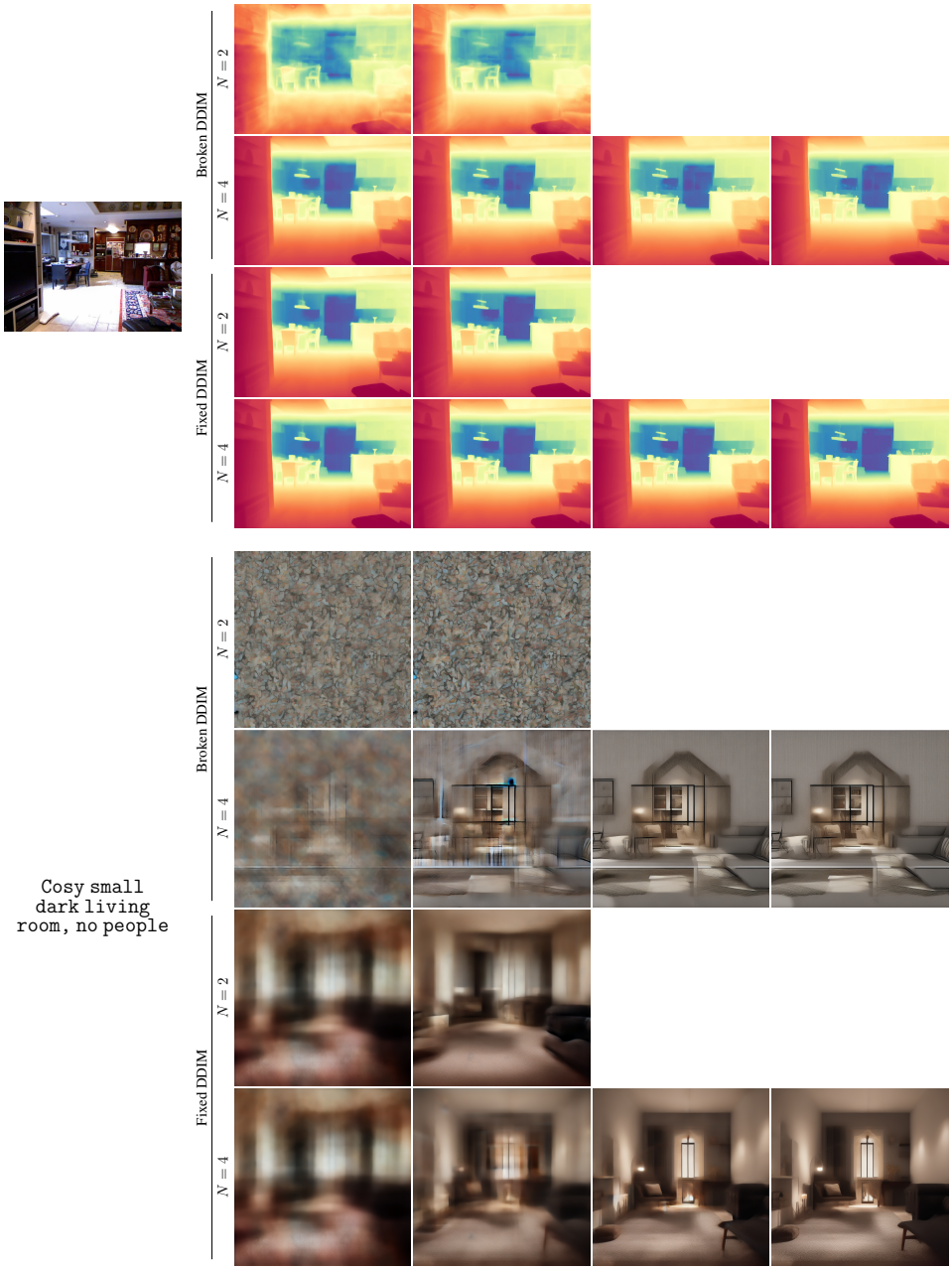

Figure 6: **Few-step inference of Marigold and Stable Diffusion.** With more steps, the adverse effects of the broken DDIM scheduler get less pronounced. Both Marigold and Stable Diffusion produce sharper outputs with more steps, but the difference is much greater for Stable Diffusion.

where $n^*_{i,j}$ is the ground-truth normal at pixel (i, j), and $\hat{n}_{i,j}$ is the predicted normal.

Training Datasets. For a direct comparison with Marigold [19], we use the same synthetic training datasets offering high quality ground-truth annotations, *i.e.*, Hypersim [28] and Virtual KITTI 2 [5].

Hypersim consists of 54 K photorealistic images from 365 indoor scenes, which we resize to a resolution of 480×640 with a far plane at 65 meters. Virtual KITTI 2 contains approximately 20 K samples from four synthetic driving scenarios under various weather conditions. These images are cropped to 352×1216 pixels, and the far plane is set to 80 meters. Following Marigold, we remove outliers, *i.e.*, values in the 2^{nd} and 98^{th} percentile, in the depth maps and normalize the remaining values to the range $[-1, 1]$. The only data augmentation is random horizontal flipping.

Since Virtual KITTI 2 does not provide normals, we estimate them from the ground-truth depth maps using discontinuity-aware gradient filters [10]. Fig. 4 shows a qualitative example of the estimated normals.

Evaluation Datasets. We evaluate the fine-tuned models on commonly used benchmarks for monocular depth estimation. NYUv2 [34] and ScanNet [6] provide RGB-D data of indoor environments captured with Kinect cameras. ETH3D [33] and DIODE [36] consist of both indoor and outdoor scenes, derived from LiDAR sensors. KITTI [12] contains outdoor driving scenes captured by vehicle-mounted cameras and LiDAR sensors. For surface normal estimation, we evaluate on NYUv2, ScanNet, and additionally on iBims-1 [20], a high-quality indoor RGB-D dataset, as well as Sintel [4], a synthetic outdoor dataset.

Evaluation Protocol. All evaluations are conducted in the zero-shot setting. We evaluate affine-invariant depth predictions using the standard approach, which involves the same scale and shift optimization between the predicted depth and the ground truth as in the loss computation [27]. Following established conventions, we report the mean absolute relative error (AbsRel), defined as the average relative difference between the ground-truth depth and the aligned predicted depth at each pixel, as well as the $\delta 1$ accuracy, which is the percentage of pixels where the ratio of the aligned predicted depth to the ground truth (and its inverse) is less than 1.25.

For surface normal predictions, we report the commonly used mean angular error (Mean) between the ground-truth normal vectors and the predictions, as well as the percentage of pixels with an angular error below 11.25 degrees.

Implementation Details. For depth estimation, we use the official Marigold checkpoint, whereas for normal estimation, we train a model with the same training setup as Marigold's depth estimation, encoding normal maps as 3D vectors in the color channels.

While a more extensive hyperparameter search might further improve results, Marigold's default hyperparameters performed well in our experiments, which we further detail below. We train all models for 20K iterations using the AdamW optimizer [23] with a base learning rate of 3×10^{-5} and an exponential learning rate decay after a 100-step warm-up. The batch size is set to 2, with gradient accumulation over 16 steps for an effective batch size of 32. This is a deliberate strategy to allow for mixing of images with different aspect ratios and resolutions. We use a specific mix of indoor and outdoor scenes from both Hypersim [28] (90%) and Virtual KITTI 2 [5] (10%), which was beneficial to the model's performance. Fine-tuning takes approximately 3 days on a single Nvidia H100 GPU.

D Additional Experimental Results

Comparison with State-of-the-Art As shown in Tab. 4, the fine-tuned models outperform current state-of-the-art generative methods for depth estimation on most datasets. Among discriminative methods, only Depth Anything [39, 40] and Metric3D [41, 17] demonstrate superior performance; however, these methods were trained on datasets that are two to three orders of magnitude larger. For surface normal estimation, the fine-tuned models set new state-of-the-art results across all evaluated datasets, with the exception of NYUv2, where Metric3D v2 continues to lead, as shown in Tab. 5.

Applying DDIM fix and End-to-End Fine-Tuning to GeoWizard. GeoWizard [11] jointly predicts depth and surface normals, using a similar training and evaluation setup as Marigold. First, we find that GeoWizard suffers from the same flaw in the DDIM implementation as Marigold; second, we find that jointly fine-tuning the model for depth and normal estimation significantly boosts the performance (see Tab. 6 and Tab. 7). In particular, the fine-tuned model performs better than both the fixed single-step model and the claimed previous best results with 50 steps and ensembling of 10 predictions.

Further Comparisons to DepthFM. DepthFM [13] proposes a direct mapping from input images to depth maps through flow matching, leveraging Stable Diffusion v2 [29] as a prior. We observe that, apart from the ETH3D $\delta 1$ and DIODE [36] metrics, a simpler approach like E2E FT achieves better performance with a more than $10\times$ speedup as seen in Tab. 8.

Fine-Tuning the VAE Decoder. By default, we keep the pretrained VAE decoder frozen while conducting end-to-end fine-tuning. Tab. 9 shows that fine-tuning the weights of this decoder does not improve performance.

Table 4: **Comparison to state-of-the-art depth estimation methods.** †Metric3D v2 [17] was trained on ScanNet, so zero-shot evaluation on this dataset is not possible. We gray out results that were not reproducible with the released code and models.

Method	Training samples	NYUv2 [34]		KITTI [12]		ETH3D [33]		ScanNet [6]		DIODE [36]	
		AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑
MiDaS [27] (TPAMI '22)	2 M	11.1	88.5	23.6	63.0	18.4	75.2	12.1	84.6	33.2	71.5
LeReS [42] (CVPR '21)	354 K	9.0	91.6	14.9	78.4	17.1	77.7	9.1	91.7	27.1	76.6
Omnidata v1 [8] (ICCV '21)	12.2 M	7.4	94.5	14.9	83.5	16.6	77.8	7.5	93.6	33.9	74.2
HDN [43] (NeurIPS '22)	300 K	6.9	94.8	11.5	86.7	12.1	83.3	8.0	93.9	24.6	78.0
DPT [26] (ICCV '21)	1.39 M	9.8	90.3	10.0	90.1	7.8	94.6	8.2	93.4	18.2	75.8
Depth Anything [39] (CVPR '24)	62 M	**4.3**	**98.1**	7.6	94.7	12.7	88.2	—	—	6.6	95.2
Depth Anything v2 [40] (arXiv '24)	62 M	4.4	97.9	7.5	94.8	13.2	86.2	—	—	**6.5**	**95.4**
Metric3D [41] (ICCV '23)	8 M	5.0	96.6	5.8	97.0	6.4	96.5	**7.4**	**94.1**	22.4	80.5
Metric3D v2 [17] (TPAMI '24)	16 M	**4.3**	**98.1**	**4.4**	**98.2**	**4.2**	**98.3**	—†	—†	13.6	89.5
Marigold [19] (CVPR '24)	74 K	5.5	96.4	9.9	91.6	6.5	96.0	6.4	95.1	30.8	77.3
GeoWizard [11] (ECCV '24)	278 K	5.2	96.6	9.7	92.1	6.4	96.1	6.1	95.3	29.7	79.2
↳ reproduced by us	278 K	5.7	96.2	14.4	82.0	7.5	94.3	6.1	95.8	31.4	77.1
DepthFM [13] (arXiv '24)	63 K	6.5	95.6	8.3	93.4	—	—	—	—	22.5	80.0
↳ reproduced by us	63 K	6.9	95.4	11.4	88.1	6.5	**96.2**	8.1	92.5	**25.0**	**78.3**
Marigold + E2E FT	74 K	**5.2**	**96.6**	**9.6**	91.9	**6.2**	95.9	**5.8**	96.2	30.2	77.9
Stable Diffusion [29] + E2E FT	74 K	5.4	96.5	**9.6**	**92.1**	6.4	95.9	**5.8**	**96.5**	30.3	77.6

Table 5: **Comparison to state-of-the-art normal estimation methods.** †Metric3D v2 [17] was trained on ScanNet, so zero-shot evaluation on this dataset is not possible. We gray out results that were not reproducible with the released code and models.

Method	Training samples	NYUv2 [34]		ScanNet [6]		iBims-1 [20]		Sintel [4]	
		Mean↓	11.25°↑	Mean↓	11.25°↑	Mean↓	11.25°↑	Mean↓	11.25°↑
Omnidata v1 [8] (ICCV '21)	12.2M	23.1	45.8	22.9	47.4	19.0	62.1	41.5	11.4
Omnidata v2 [18] (CVPR '22)	12.2M	17.2	55.5	**16.2**	60.2	18.2	63.9	40.5	14.7
DSINE [2] (CVPR '24)	161K	16.4	59.6	**16.2**	**61.0**	**17.1**	67.4	**34.9**	**21.5**
Metric3D v2 [17] (TPAMI '24)	16M	**13.3**	**66.4**	—†	—†	19.6	**69.7**	—	—
Marigold [19] (CVPR '24)	74K	18.8	55.9	17.7	58.8	18.4	64.3	39.1	14.9
GeoWizard [11] (ECCV '24)	278K	17.0	56.5	15.4	61.6	13.0	65.3	—	—
↳ reproduced by us	278K	19.1	49.5	17.3	53.7	19.5	61.6	40.4	13.2
GeoWizard + E2E FT	278K	**16.1**	60.7	15.3	63.6	16.2	69.4	**33.4**	**22.4**
Marigold + E2E FT	74K	16.2	**61.4**	**14.7**	66.0	**15.8**	**69.9**	33.5	21.5
Stable Diffusion [29] + E2E FT	74K	16.5	60.4	**14.7**	**66.1**	16.1	69.7	33.5	22.3

Deterministic or Probabilistic. We perform an ablation on the type of noise used during fixed-timestep fine-tuning; the results are shown in Tab. 10. "Gaussian" and "Pyramid" refer to the standard normal and multi-resolution noise commonly employed in diffusion training and used in Marigold, respectively. "Zeros" describes our default setting, *i.e.*, no noise. We find that using constant zeros performs slightly better than the alternatives, although the method seems to be fairly robust to the actual choice of noise.

Qualitative Results. Fig. 7 and Fig. 8 show qualitative results for depth and normals estimation, respectively, comparing Marigold [19] and the end-to-end fine-tuned models. The fixed single-step model fails to produce sharp results, while the multi-step model exhibits noticeable over-sharpening and high-frequency noise artifacts (even after ensembling), particularly in the normals estimations. In contrast, the end-to-end fine-tuned models do not exhibit these issues.

Table 6: **Fixed DDIM scheduler and end-to-end fine-tuning (E2E FT) for GeoWizard's [11] depth estimation.** We use the official code and model weights to re-evaluate the method on all datasets. Inference time is for a single 576×768-pixel image, evaluated on an NVIDIA RTX 4090 GPU. We obtain significant speed-ups, improving results.

Method	Steps	Ens.	Inference time	NYUv2 [34]		KITTI [12]		ETH3D [33]		ScanNet [6]		DIODE [36]	
				AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑
GeoWizard [11]	50	10	72 s	5.2	96.6	9.7	92.1	6.4	96.1	6.1	95.3	29.7	79.2
↳ reproduced by us	50	10	72 s	5.7	**96.2**	14.4	82.0	7.5	94.3	6.1	95.8	31.4	77.1
GeoWizard + DDIM fix	1	1	**254** ms	5.8	96.1	13.3	84.7	7.8	94.3	6.2	95.7	32.0	76.0
GeoWizard + E2E FT	1	1	**254** ms	**5.6**	96.1	**9.8**	**91.4**	**6.3**	**95.7**	**5.9**	**96.2**	**30.6**	**77.9**

Table 7: **Fixed DDIM scheduler and end-to-end fine-tuning (E2E FT) for GeoWizard's [11] normal estimation.** We use the official code and model weights to re-evaluate the method on all datasets. Inference time is for a single 576×768-pixel image, evaluated on an NVIDIA RTX 4090 GPU. We obtain significant speed-ups, improving results. GeoWizard's original results include additional post-processing steps, such as smoothing.

Method	Steps	Ens.	Inference time	NYUv2 [34]		ScanNet [6]		iBims-1 [20]		Sintel [4]	
				Mean↓	11.25°↑	Mean↓	11.25°↑	Mean↓	11.25°↑	Mean↓	11.25°↑
GeoWizard [11] (ECCV 24)	50	10	72 s	17.0	56.5	15.4	61.6	13.0	65.3	—	—
↳ reproduced by us	50	10	72 s	19.1	49.5	17.3	53.7	19.5	61.6	40.4	13.2
GeoWizard + DDIM fix	1	1	**254** ms	17.0	54.1	15.5	59.3	18.3	62.5	35.9	15.6
GeoWizard + E2E FT	1	1	**254** ms	**16.1**	**60.7**	**15.3**	**63.6**	**16.2**	**69.4**	**33.4**	**22.4**

Table 8: **Comparison of DepthFM [13] with the DDIM-fixed and end-to-end fine-tuned (E2E FT) Marigold and Stable Diffusion models.** We re-evaluated DepthFM [13] on all datasets using the official code and model weights, with 4 inference steps and an ensemble size of 6. Inference time is for a single 576×768-pixel image, evaluated on an NVIDIA RTX 4090 GPU.

Method	Steps	Ens.	Inference time	NYUv2 [34]		KITTI [12]		ETH3D [33]		ScanNet [6]		DIODE [36]	
				AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑
DepthFM [13]	4	6	1.67 s	6.5	95.6	8.3	93.4	—	—	—	—	22.5	80.0
↳ reproduced by us	4	6	1.67 s	6.9	95.4	11.4	88.1	6.5	**96.2**	8.1	92.5	**25.0**	78.3
DepthFM	1	1	132 ms	7.5	95.0	11.6	87.5	6.7	96.0	8.3	92.3	25.3	**77.9**
Marigold [19] + E2E FT	1	1	**121** ms	**5.2**	**96.6**	**9.6**	91.9	**6.2**	95.9	**5.8**	96.2	30.2	**77.9**
SD [29] + E2E FT	1	1	**121** ms	5.4	96.5	**9.6**	**92.1**	6.4	95.9	**5.8**	**96.5**	30.3	77.6

Table 9: **Frozen vs. fine-tuned VAE decoder.** We conduct end-to-end fine-tuning of Marigold [19] for depth estimation, and assess the effect of freezing or fine-tuning the weights of the pretrained VAE decoder.

Decoder	NYUv2 [34]		KITTI [12]		ETH3D [33]		ScanNet [6]		DIODE [36]	
	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑
Frozen	**5.2**	**96.6**	**9.6**	**91.9**	**6.2**	95.9	**5.8**	**96.2**	**30.2**	**77.9**
Fine-tuned	5.3	96.5	**9.6**	**91.9**	**6.2**	**96.0**	**5.8**	96.1	**30.2**	77.7

Table 10: **Deterministic or probabilistic.** The effect of different types of noise for task-specific fine-tuning for depth estimation.

Noise	NYUv2 [34]		KITTI [12]		ETH3D [33]		ScanNet [6]		DIODE [36]	
	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑
Marigold [19] fine-tuning										
Gaussian	5.3	96.4	9.9	91.4	6.3	95.9	5.9	96.0	30.5	77.3
Pyramid	5.4	96.5	9.9	91.0	6.3	**96.0**	6.0	95.9	**30.1**	77.7
Zeros	**5.2**	**96.6**	**9.6**	**91.9**	**6.2**	95.9	**5.8**	**96.2**	30.2	**77.9**
Stable Diffusion [29] fine-tuning										
Gaussian	5.8	96.1	9.8	91.5	6.6	95.5	6.0	96.1	30.7	77.2
Zeros	**5.4**	**96.5**	**9.6**	**92.1**	**6.4**	**95.9**	**5.8**	**96.5**	**30.3**	**77.6**

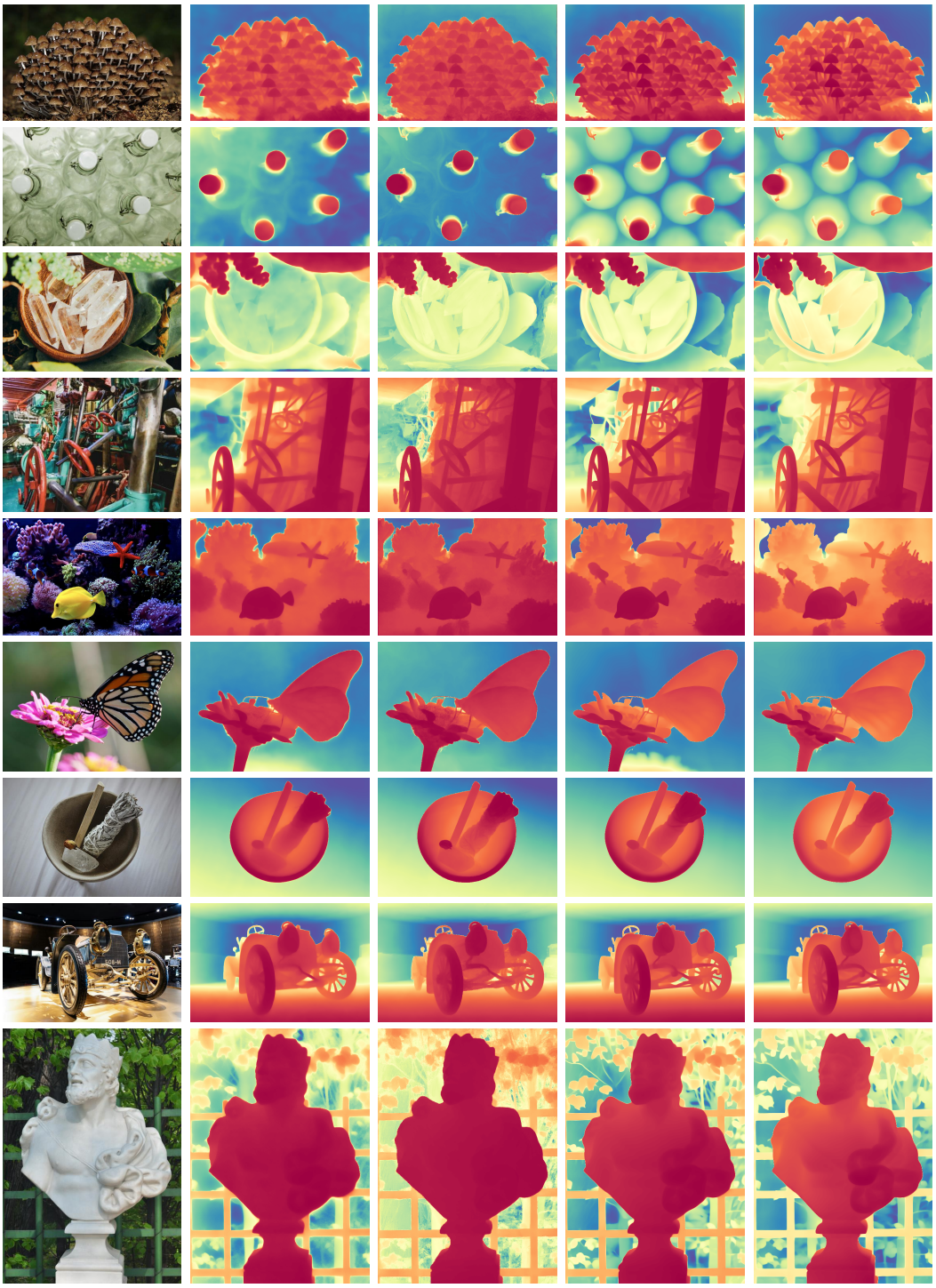

Figure 7: **Qualitative results for depth estimation.** "Marigold (X, Y)" denotes Marigold using X inference steps with an ensemble of size Y.

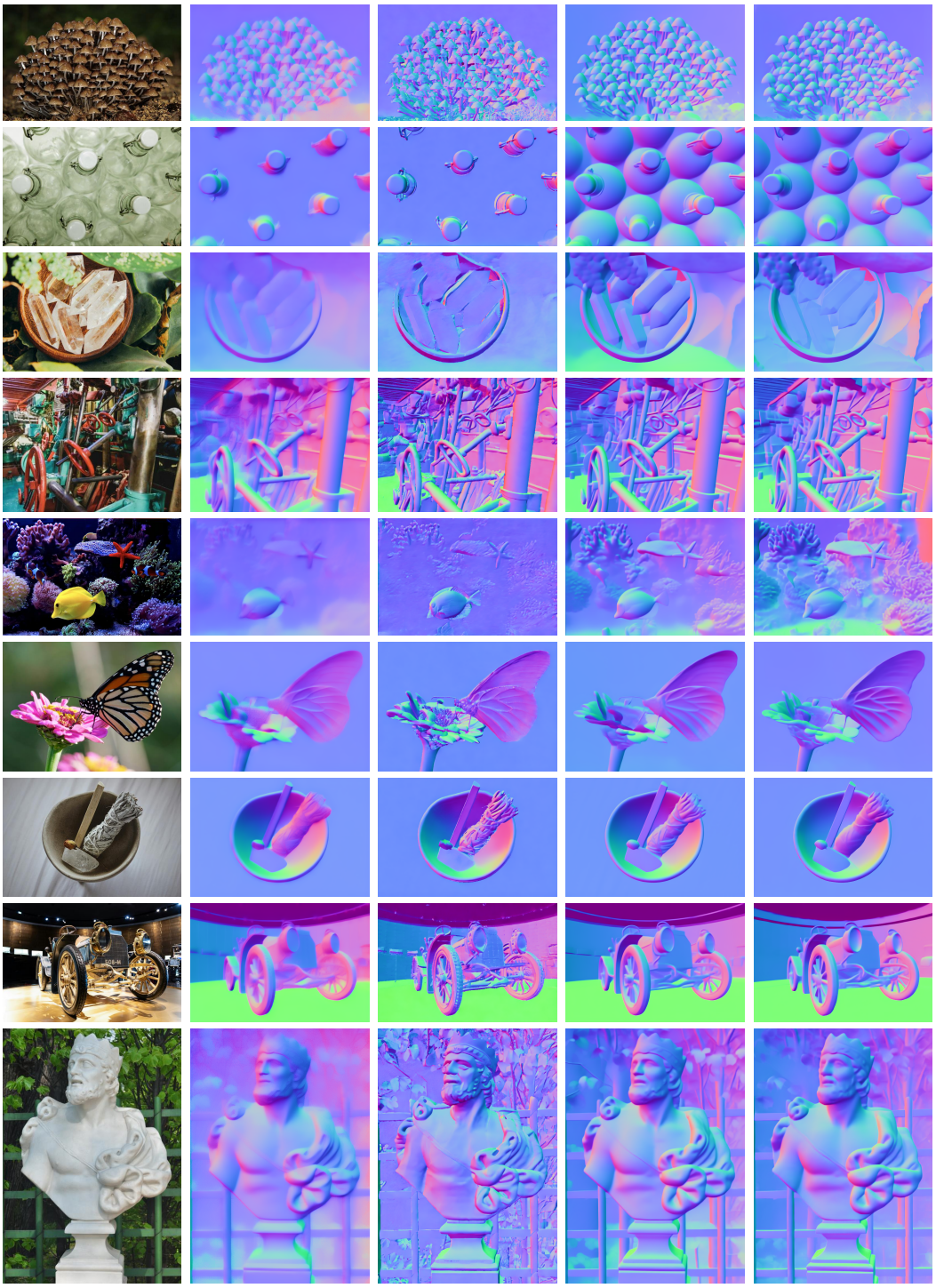

Figure 8: **Qualitative results for normal estimation.** "Marigold (X, Y)" denotes Marigold using X inference steps with an ensemble of size Y.